REMARKS/ARGUMENTS

This is in response to an Office action dated December 11, 2006 having a statutory period of response set to expire on March 11, 2006.

Claim Rejections -35 USC § 112

Claims 1-15, 21 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential elements, such omission amounting to a gap between the elements. The omitted elements are: first limitation of claims 1, 14, 21 is formulated unclear: what is "least enclosing rectangle enclosing a conductor region shape" and how it's related to "contact shapes" and where this shapes are located? For examination purposes Examiner considers that all shapes representing the components of the integrated circuit layout are in rectangular shapes.

Claims 1, 14 and 21 have been amended to claim "computing a rectangular shape that encloses a conductor region shape and contact shapes on the integrated circuit design, said rectangle shape being the smallest rectangular shape that completely encloses the conductor region shape and the contact shapes..." It is now clear that the "rectangular shape" as now defined " encloses a conductor region shape and contact shapes on the integrated circuit design.." In addition, the rectangle shape is further defined as "being the smallest rectangular shape that completely encloses the conductor region shape and the contact shapes." This language is supported by paragraph 44, in the application as originally filed and therefore does not include any new matter, Accordingly, the 112 rejection of claims 1, 14 and 21 should be withdrawn.

Claim Rejections -35 USC § 102

Claims 1-2, 4-13 are rejected under 35 U.S.C. 102(b) as being anticipated by Bhat et al. ("Special Purpose Architecture for Accelerating Bitmap DRC", 25-29 June 1989, Design Automation, 26th Conference on, Pages: 674 - 677).

With respect to claim 1 Bhat et al. teaches a method of performing latch up check on an integrated circuit (IC) design (within performing design rule check specifically for width and space checking on design integrated circuit) (abstract, Page 674) comprising the steps of: computing a combined least enclosing rectangle enclosing a conductor region shape and contact

shapes (by determining rectangular geometries in the layout of the design IC) (Page 674, left column); rasterizing the conductor region shape and the contact shapes (using rasterized layout as a bitmap pattern, such as pixel array) (Page 674, left column); iteratively expanding the contact shapes within the conductor region shape using a cellular algorithm (within a window flexible in size using a concept of shrink/expand instructions iteratively processing rectangular geometries using cellular algorithm as manipulating with grid of cells (pixels) in subsections or windows (contact shapes) with very small instruction set) (Page 674, left, right columns); generating shapes representing an unreachable area of the conductor region shape (using algorithm for design rule check (DRC) for generating the check layer for checking rule violations in four directions — horizontal (left to right), vertical (top to bottom), along the 45° direction (top to bottom) and along the 135° direction (top to bottom) including spacing checks based on bitmap layout (Page 675, left column); and checking the shapes representing the unreachable area of the conductor region shape against junction shapes in the design (within generating check layer and performing layout analysis for determining space rule violation as shown on the Fig. 3.1 representing check layer) (Page 675, left column), and reporting to a designer any junction shapes which intersect the **unreachable area as errors** within generating the error layer as shown on the Fig. 3.2 containing error positions corresponding to the direction in which checking was performed showing l's for error and U's for no error (Page 675, left and right columns), which is obtaining error locations (unreachable areas) (Page 675, right column).

With respect to claims 2-13 Bhat et al. teaches:

Claim 2: including the steps of: representing the contact shapes as cells in a byte array (as shown on the Fig. 2.1 depicting a bitmap representation of a single layer in the layout polygons as a array of pixels) (Page 674, right column); and exploring the conductor region shape by expanding the conductor region shape into neighboring cells of the byte array (within window processor architecture for implementation of rasterized layout shown on the Fig. 2.1 using shrink/expand instructions to perform DRC design IC) (Page 674, left column);

Claim 4: further including the step of restricting the number of directions in which a cell can expand (associating the number of directions for expanding with the number of directions of

checking cells on the check layer) (Page 675, left column);

Claim 5: further including the step of creating a 2-dimensional byte array of sufficient size to rasterize the enclosing rectangle at a resolution of "I" (within the widow processor across the entire layout checking if the bitmap pattern within the window is valid, and each pixel of the bitmap is processed n^2 for a window of size n x n) (Page 674, left column), wherein the width and height of each cell in the array corresponds to the value "I" (within each pixel as elementary square having length and width = ?, where ? is the unit in which the design rules are expressed) (Page 674, right column);

Claim 6: including the step of initializing each cell of the byte array to a first code representing an empty cell (within white pixels (empty) if it does not describe any layout polygon) (Page 674, right column);

Claim 7: further including the step of converting the least enclosing rectangle to raster format in the byte array by inserting a second code into each cell intersected by an edge of the rectangle shape (within black pixels which form a part of region within any layout poilygon) (Page 674, right column);

Claim 8: further comprising the step of converting the conductor region shape to raster format in the byte array by inserting a third code into each cell intersected by an edge of the conductor region shape (within edge pixels, which are black pixels and form the polygon boundary) (Page 674, right column);

Claim 9: further comprising the steps of: converting the contact shapes to raster format in the byte array by inserting a fourth code into each cell intersected by an edge of a contact shape (within generating a concave corner pixels showing two edges forming a part of a layout polygon boundary and convex corner pixels showing no edges forming a part of a polygon boundary) (Page 674, right column; Page 675, right column); and recording the address of each of these cells in a frontier list (within a cell shown on the Fig. 4.1, which is one of components of the processing elements construing design rule checker chip, wherein cell contain a register for

storing bitmap data of each pixel including neighbor information) (Page 676, left column);

Claim 10: further comprising the step of establishing a maximum distance to be searched (within limited of the length of the maximum rule that can be checked, and is 4 ?) (Page 675, right column; Page 676, right column);

Claim 11: further comprising the steps of: expanding the contact shapes by traversing the frontier list one cell at a time and examining the cell's neighbor cells as to whether they are empty or not (within shrink/expand operations, which involves checking the neighbor pixels and determining if they empty (white), making the pixel under checking is made white with the consideration of the length of the maximum rule using the register of the cells shown on the Fig. 4.1 for storing bitmap data including neighbor information for use during shrink/expand operations) (Page 675, right column; Page 676, left column); inserting a fifth into the neighbor cell and recording its location in a new frontier list if a neighbor cell is empty (within checking the neighbor pixels and determining if they empty (white), making the pixel under checking is made white) (Page 675, right column); and not expanding into it if a neighbor cell is not empty (using rule aligners to generate control signals for plane modules shown on the Fig. 4.1, necessary hardware to combine neighbor information for use shrink/expand operations) (Page 676, left column);

Claim 12: further comprising the steps of: expanding cells which are recorded in the new frontier list (within shrink/expand operations, which involves checking the neighbor pixels and determining if they empty (white), making the pixel under checking is made white with the consideration of the length of the maximum rule using the register of the cells shown on the Fig. 4.1 for storing bitmap data including neighbor information for use during shrink/expand operations, wherein iteration is available up to 4 times) (Page 675, right column; Page 676, left column; Page 674, right column); and inserting a fifth code into the neighbor cell and recording its location in the new frontier list if a neighbor cell is empty (within checking the neighbor pixels and determining if they empty (white), making the pixel under checking is made white, wherein iteration is available up to 4 times) (Page 675, right column; Page 674, right column); and not expanding into it if a neighbor cell is not empty (using rule aligners to generate control

signals for plane modules shown on the Fig. 4.1, necessary hardware to combine neighbor information for use shrink/expand operations, wherein iteration is available up to 4 times) (Page 676, left column; Page 674, right column);

Claim 13: further comprising the steps of: continuing to expand cells by traversing the new frontier list one cell at a time, and examining the cell's neighbor cells (within iteration which is available: up to 4 times) (Page 674, right column); and inserting a sixth code into the cell, and record its location in a third frontier list if a neighbor cell is empty (within 'A' register to store bitmap data of pixels after shrink/expand operations) (Page 675, left column).

Claim 1 has been amended to incorporate the limitations of claims 2 and 3 and therefore the rejection of claims 1, 2 and 4-13 is moot.

Since claim 1 was objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims, claim 1 should be deemed allowable.

Allowable Subject Matter

Claims 16-20 were allowed.

Claims 14, 15 and 21 have allowable subject matter similar to aforementioned claims 16-20. Claims 14, 15 and 21 might be allowed after they overcome rejection under 35 USC § 112 second paragraph.

Claim 14 has been amended to overcome the 112 rejection and should be deemed allowable.

Claim 15 depends on claim 14 and should be allowable.

Claim 21 has been amended to overcome the 112 rejection and should be deemed allowable.

Claim 3 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Claim 1 has been amended to incorporate the limitations of claims 2 and 3 and therefore claim 1 is allowable.

Claim 4 has been amended to depend upon claim 1 and should also be deemed allowable.

Claim 5 depends on claim 1 and should be allowable.

Claims 6-13 depend upon claim 5 and should also be deemed allowable.

Conclusion

Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

If here are any issues that still need to be resolved, the Examiner is invited to contact the undersigned Attorney of record.

Respectfully submitted,

Howard M. Cohn
Registration No. 25,808

Howard M. Cohn
21625 Chagrin Blvd. Suite 220
Cleveland, OH 44122
Voice (216) 752-0955
Fax (216) 752-0957

C:\Documents and Settings\Office\My Documents\IBM\IBM-112\ibm-112 2am.doc